ปราโมทย์ พลังสันติกุล

Visual Basic : Communication custom control (MSComm)

แนะนำ Control ที่ใช้ในการติดต่อกับ RS-232 ใน Visual Basic เรียกว่า MSComm Control

จากการที่ผู้เขียนไม่ได้ศึกษามาทางด้านคอมพิวเตอร์โดยตรง และต้องทำงานที่เกี่ยวกับการเขียนโปรแกรมคอมพิวเตอร์เพื่อติดต่อกับอุปกรณ์ภายนอก คือ Micro Controller ซึ่งจะต้องเขียนโปรแกรมเพื่อทำการติดต่อกับอุปกรณ์ภายนอกนั้น ก็คงหนีไม่พ้นที่จะต้องใช้ RS-232 (Serial Communication Device)ในการติดต่อ

การพัฒนาโปรแกรมในส่วนของคอมพิวเตอร์ ผู้เขียนเลือกใช้ Visual Basic ในการจัดทำโครงงานดังกล่าว เพราะคิดว่า Visual Basic ใช้เวลาในการเรียนรู้ไม่นานก็สามารถที่จะพัฒนาโปรแกรมใช้งานได้ไม่ยาก จากการได้ศึกษา Visual Basic นั้นพบว่า จะมีข้อมูลน้อยมากที่เกี่ยวกับการเขียนโปรแกรมติดต่อกับ RS-232 โดยเฉพาะหนังสือภาษาไทยที่เกี่ยวกับการเขียนโปรแกรมด้วย Visual Basic ทางด้าน Serial Communication นั้น อาจหาไม่ได้เลย

นอกจากนี้ผู้เขียนได้ให้คำแนะนำเกี่ยวกับการเขียนโปรแกรมติดต่อกับ RS-232 กับเพื่อนและรุ่นน้องที่ต้องทำโครงงานแบบเดียวกับผู้เขียนเสมอ ผู้เขียนจึงมีแนวความคิดว่าน่าจะเสนอแนวทางการใช้ Visual Basic ที่เกี่ยวข้องกับ RS-232 บ้าง เพื่ออาจจะเป็นประโยชน์กับผู้ที่ต้องการใช้ Visual Basic ในการพัฒนาโปรแกรมที่ติดต่อกับ RS-232 หรือกับอุปกรณ์ภายนอกที่จะต้องติดต่อโดยผ่านทาง RS-232

บทความนี้ผู้เขียนจะแนะนำ Control ที่ใช้ในการติดต่อกับ RS-232 ใน Visual Basic เรียกว่า MSComm Control (Communication Custom Control) ส่วนข้อมูลทางด้าน Serial Port (Serial Communication Device (RS-232)) หรือ Com Port ผู้อ่านจะหาข้อมูลจากหนังสือที่เกี่ยวกับ Serial Port ได้ไม่ยาก ที่ร้านหนังสือ SE-ED BOOK CENTER ก็มีหลายเล่ม เลือกหาได้ตามใจชอบเลยนะครับ

การเขียนโปรแกรมสื่อสารผ่านทาง Serial Port ด้วย Visual Basic นั้น สิ่งที่ต้องเรียนรู้ประกอบไปด้วยรายละเอียดดังต่อไปนี้

1. ต้องมี MSCom Control ซึ่งก็จะมีอยู่แล้วเมื่อทำการติดตั้ง Visual Basic

2. กำหนด Custom Control MSCOMM.OCX (สำหรับ Visual Basic 4 ขึ้นไป) โดยการเลือก MSCOMM.OCX จากเมนู Project -> Components -> ที่ Control เลือก Microsoft Comm Control 5.0 (ผู้เขียนได้ใช้ Visual Basic 5) จะได้ Control MSComm (รูปโทรศัพท์) ปรากฏในส่วนของ ToolBox

3. นำ Control MSCOMM ไปไว้ใน Form ที่ต้องการใช้งาน

4. กำหนด Properties ให้กับ Control เขียน function ที่เกี่ยวข้องกับ MSCOMM

MSCOMM Control ใช้สำหรับการติดต่อกับ Serial Port ทั้งการรับและการส่งข้อมูลให้กับ Serial Port ซึ่งจะมีการติดต่อได้ 2 แบบ คือ

1. การสื่อสารแบบกระตุ้นด้วยเหตุการณ์ (Event-driven communications) หรือเป็นการเขียนโปรแกรมแบบกระตุ้นด้วยอินเทอร์รัพต์นั่นเอง โดยเมื่อมีข้อมูลเข้ามาก็จะเกิด CommEvent ขึ้นกับ OnComm Events

2. การสื่อสารแบบโพลลิ่ง (Polling) โดยการวนรอบตรวจสอบข้อมูลจาก Serial Port ตลอดเวลา

MSCOMM Control ประกอบไปด้วย Events และ Properties ดังต่อไปนี้

1. MSCOMM CONTROL EVENTS

MSCOMM Control จะมี Events เพียง Events เดียวเท่านั้น คือ OnComm Events ซึ่งประกอบไปด้วย CommEvent property ดังต่อไปนี้

- เมื่อเกิดเหตุผิดพลาดในการสื่อสาร จะแสดงใน CommEvent property ดังตารางที่ 1
- เมื่อเกิดสถานะการสื่อสารขึ้น จะแสดงใน CommEvent property ดังตารางที่ 2

ซึ่ง comEvent ดังกล่าวจะเกิดขึ้นตามสถานะการสื่อสาร ว่าเราได้กำหนดไว้อย่างไร ซึ่งหากเป็นการเขียนโปรแกรมติดต่อสื่อสารแบบธรรมดาก็จะใช้เพียงไม่กี่ comEvents เท่านั้น เช่น comEvReceive, ComEvSend ส่วนที่แสดงทั้งหมดนั้นจะใช้ก็ต่อเมื่อเราได้กำหนดให้มีเงื่อนไขมากขึ้นในการตรวจสอบสายสัญญาณ โดยเฉพาะการสื่อสารผ่านทางโมเด็ม

ตารางที่ 1

CommEvent Property	ความหมาย
ComEventBreak	ได้รับสัญญาณเบรก
ComEventCDTO	เกิดการไทม์เอาต์ ขณะที่กำลังรอสัญญาณ CD (Carrier Detect)
ComEventCTSTO	เกิดการไทม์เอาต์ ขณะที่กำลังรอสัญญาณ CTS (Clear To Send)
ComEventDSRTO	เกิดการไทม์เอาต์ ขณะที่กำลังรอสัญญาณ DSR (Data Set Ready)
ComEventFrame	เกิดความผิดพลาดทางเฟรม คือ ไม่พบบิตจบตามที่ควรจะเป็น
ComEventOverrun	เกิดความผิดพลาดโอเวอร์รัน คือ การที่รับข้อมูลไม่ทันในการประมวลผล
ComEventRxOver	บัฟเฟอร์รับข้อมูลเกิดโอเวอร์โฟวล์หรือรับตัวอักษรหลังจากการรับ EOF Char
ComEventRxParity	เกิดความผิดพลาดทางพาริตี้ คือ ตัวอักษรที่รับได้มีบิตพาริตี้ไม่ถูกต้อง
ComEventTxFull	บัฟเฟอร์ส่งข้อมูลเต็ม
ComEventDCB	ความผิดพลาดที่เกิดขึ้นโดยไม่ได้คาดถึง

ตารางที่ 2

CommEvent Property	ความหมาย
ComEvCD	CD (Carrier Detect) เปลี่ยนสถานะ ซึ่งก็คือสายสัญญาณ Receive Line Signal Detect (RLSD)
ComEvCTS	CTS (Clear To Send) มีการเปลี่ยนสถานะเกิดขึ้น
ComEvDSR	DSR (Data Set Ready) มีการเปลี่ยนสถานะเกิดขึ้น
ComEvRing	ตรวจจับสัญญาณ Ring Indicator ได้
ComEvReceive	ได้รับข้อมูลเก็บลง InputBuffer
ComEvSend	ส่งข้อมูลออกจาก OutputBuffer
ComEvEof	พบอักขระ EOF

2. MSCOMM CONTROL PROPERTY

MSCOMM Control Property ประกอบไปด้วย property ดังตารางที่ 3

Property	ความหมาย
Break	กำหนดหรือเคลียร์สัญญาณเบรก
CDHolding	ตรวจสอบสัญญาณ Carrier Detect (CD) ว่ายังคงมีสถานะอยู่หรือไม่
CDTimeout	กำหนดค่าหรือให้ค่าของเวลา (หน่วย มิลลิวินาที) ที่รอสัญญาณ Carrier Detect (CD)
CommEvent	ให้ผลของการเกิด Event ของ Communucation
CommID	ให้ผลของเฮลเดิลของ Communication ที่เปิด
CommPort	กำหนดหรือให้ผลของหมายเลขคอมพอร์ตที่เปิด
CTSHolding	ตรวจสอบสัญญาณ Clear To Send (CTS) ว่ายังคงมีสถานะอยู่หรือไม่
CTSTimeout	กำหนดค่าหรือให้ค่าของเวลา (หน่วย มิลลิวินาที) ที่รอสัญญาณ Clear To Send (CTS)
DSRHolding	ตรวจสอบสัญญาณ Data Set Ready (DSR) ว่ายังคงมีสถานะอยู่หรือไม่
DSRTimeout	กำหนดค่าหรือให้ค่าของเวลา (หน่วย มิลลิวินาที) ที่รอสัญญาณ Data Set Ready (DSR)
DTREnable	อีนาเบิล สายสัญญาณ Data Terminal Ready (DTR)
Handshaking	กำหนดการแฮนเช็คทางฮาร์ดแวร์ เพื่อคอยตรวจสอบการรับส่งข้อมูล
InBufferCount	ให้ผลของจำนวนข้อมูลที่อยู่ในบัฟเฟอร์รับข้อมูล
InBufferSize	กำหนดหรือให้ผลของขนาดบัฟเฟอร์รับข้อมูล
Input	ให้ผลหรือย้ายข้อมูลจากบัฟเฟอร์รับข้อมูล
InputLen	กำหนดหรือให้ผลของจำนวนข้อมูลที่จะนำมาจากบัฟเฟอร์รับข้อมูล
Interval	กำหนดอัตราการความเร็วสำหรับการใช้งานในโหมดโพลลิ่ง
NullDiscard	กำหนดให้มีการรับ Null Character เก็บลงบัฟเฟอร์รับข้อมูล
OutBufferCount	จำนวนข้อมูลที่รออยู่ในบัฟเฟอร์ส่งข้อมูล
OutBufferSize	กำหนดหรือให้ผลขนาดของบัฟเฟอร์ส่งข้อมูล
Output	ส่งข้อมูลให้กับบัฟเฟอร์ส่งข้อมูลเพื่อส่งออก
ParityReplace	กำหนดให้ส่งอักขระนี้แทนหากเกิดการผิดพลาดในข้อมูล
PortOpen	กำหนดหรือให้ผลของสถานะพอร์ตว่าเปิดหรือปิดอยู่
Rthreshold	กำหนดหรือให้ผลของจำนวนข้อมูลที่เก็บลงบัฟเฟอร์รับข้อมูลก่อนการเกิด CommEvent ในการรับข้อมูล
RTSEnable	อีนาเบิล สัญญาณ Request To Send (RTS)
Settings	กำหนดอัตราบอด พาริตี้ ข้อมูล บิตหยุด
Sthreshold	กำหนดหรือให้ผลของจำนวนข้อมูลที่เก็บลงบัฟเฟอร์ส่งข้อมูลก่อนการเกิด CommEvent ในการส่งข้อมูล

ตารางที่ 3

ซึ่งรายละเอียดทั้งหมดของ MSComm Control ท่านสามารถหาอ่านได้จาก Help ใน Visual Basic ในส่วนของ MSComm Control

ที่ผู้เขียนกล่าวมาทั้งหมดมีส่วนสำคัญที่ผู้ใช้งานควรที่จะรู้ ดังต่อไปนี้คือ

1. General ประกอบไปด้วย

ComPort การกำหนดหมายเลขพอร์ตในการติดต่อ RS-232 เช่น Com1, Com2 เป็นต้น

Settings การกำหนดอัตราบอด พาริตี้ จำนวนข้อมูล บิตจบ เช่น 9600, n, 8, 1

HandShaking กำหนดได้ 4 แบบ คือ 1.ComNone 2. ComXonXoff 3. ComRTS 4. ComRTSXonXoff

2. Buffer ประกอบไปด้วย

InBufferSize กำหนดบัฟเฟอร์ในการรับข้อมูล

OutBufferSize กำหนดบัฟเฟอร์ในการส่งข้อมูล

Rthreshold กำหนดการเกิด Event-driven ในการรับข้อมูล

Sthreshold กำหนดการเกิด Event-driven ในการส่งข้อมูล

Inputlen จำนวนขนาดข้อมูลที่จะอ่านมาจากบัฟเฟอร์รับข้อมูล

EOFEnable End Of File (EOF)

3. HardWare

ParityReplace อักขระที่จะแทนเมื่อเกิด Parity Error

NullDiscard กำหนดว่าจะรับหรือไม่รับ NULL CHARACTER

RTSEnable อีนาเบิล สัญญาณ Request To Send (RTS)

DTREnable อีนาเบิล สัญญาณ Data Terminal Ready (DTR)

สิ่งสำคัญที่ต้องกำหนดเพื่อให้ MSComm Control ทำงานได้ คือ

1. กำหนด ComPort โดยใช้ CommPort Property เช่น MSComm1.CommPort =1 ใช้ Com1

2. กำหนด Settings โดยใช้ Settings Property เช่น MSComm1.Settings = "9600,N,8,1" กำหนดให้ 9600 baud, no parity, 8 data and 1 stop bit

3. กำหนด InputLen โดยใช้ InputLen Property เช่น MSComm1.InputLen = 0 คือ ให้มีการอ่านข้อมูลมาทั้งหมดในบัฟเฟอร์เมื่อมีการใช้ Input Property

4. กำหนด PortOpen โดยใช้ PortOpen Property เช่น MSComm1.PortOpen = True เป็นการเปิด Port ใช้งาน

5. กำหนด Rthreshold โดยใช้ Rthreshold Property เมื่อต้องการใช้งาน Event-driven เช่น MSComm1. Rthreshold = 1 ให้เกิด Event-driven ทุกครั้งเมื่อมีข้อมูลในบัฟเฟอร์รับข้อมูล

อธิบายโปรแกรมการทำงานของโพรซีเดอร์

ตัวแปรทั่วไป

Dim ModeRXTX As Integer
กำหนดเพื่อใช้ในการสลับโหมดการทำงาน

Dim ChkStopLoopUnLess As Boolean
กำหนดใช้สำหรับการออกจากการวนรอบ

Public Echo As Boolean
กำหนดใช้แสดงข้อมูลในการป้อนค่าในส่วนแสดงข้อมูลว่าจะให้แสดงหรือไม่

Sub ที่กำหนดใน VBCom.Frm ชื่อ frmVBCom

Private Sub Form_Load()

โพรซีเดอร์ ในส่วน Form Load เป็นการโหลดค่าต่าง ๆ ที่ต้องใช้เริ่มต้นการทำงานของโปรแกรม VBCom เช่น การกำหนด Baud, Com Port, Handshaking ซึ่งเป็นส่วนสำคัญในการเริ่มเปิดการใช้งาน Serial Port โดยจะอ่านมาจาก Registry ของระบบวินโดวส์หลังจากเราได้กำหนดแล้วในปุ่ม SetCom เช่น

```
Settings = GetSetting(App.Title, "Properties",
"Settings", "") เป็นการอ่านค่า Setting ว่ากำหนดอย่างไร
เช่น 2400,8n,1
CommPort = GetSetting(App.Title, "Properties",
"CommPort", "") เป็นการอ่านค่า Com Port ว่าเป็น Com Port
ที่เท่าไหร่ เป็นต้น
```

Private Sub cmdOnOffPort_Click()

โพรซีเดอร์ cmdOnOffPort_Click เป็นส่วนการเปิดและปิด Com Port เพื่อใช้งาน

Private Sub UnLessLoop()

โพรซีเดอร์ UnLessLoop เป็นส่วนของการวนรอบตลอดเวลา โดยจะมี 2 ส่วน คือ 1. การทำงานในโหมด Event-driven และ 2. การทำงานโหมด Polling โดยการเลือกโหมดการทำงาน ซึ่งในระหว่างการทำงานนั้นสามารถสลับโหมดการทำงานได้ทันทีในขณะโปรแกรมทำงานอยู่ จะสังเกตเห็นได้ว่าในส่วนของโหมด Event-driven จะไม่มีส่วนของการรับหรือแสดงข้อมูลเป็นการวนรอบอย่างเดียว ส่วนโหมด Polling จะมีส่วนของการรับข้อมูลและแสดงข้อมูลอยู่ในการวนรอบด้วย

ในการรับข้อมูลจะใช้ Input Property ดังตัวอย่างคือ Buffer = MSComm1.Input เมื่อมีการสลับโหมดการทำงาน ก็จะไปอีกโหมดหนึ่งด้วยคำสั่ง GOTO ซึ่งจะตรวจสอบค่า ModeRXTX ว่าเป็นอย่างไร คือ หากเป็น 0 จะกระโดดไปทำงานในส่วนโหมด Event-Driven แต่หากเป็น 1 จะไปทำงานในส่วนของโหมด Polling ดังตัวอย่างนี้

```
If ModeRXTX = 0 Then GoTo MEVENT_DRIVEN
'goto mode Event-Driven
If ModeRXTX = 1 Then GoTo MPOLLING '
goto mode Polling
```

Private Sub optModeRXTX_Click (Index As Integer)

โพรซีเดอร์ optModeRXTX_Click(Index As Integer) ใช้สำหรับการสลับโหมดการทำงานของโปรแกรม โดยการกำหนดค่าให้กับ Property Rthreshold ซึ่งโปรแกรมตัวอย่างนี้ จะกำหนดให้เป็น 1 ซึ่งก็คือจะเกิด Events comEvReceive ทุกครั้งใน Sub MSComm1_OnComm() เมื่อมีข้อมูลในบัฟเฟอร์ รับข้อมูล และหากกำหนดค่า Property Rthreshold ให้เป็น 0 ก็จะไม่เกิด Event-driven ขึ้น Property นี้สำคัญมาก เพราะเป็นการกำหนดว่าจะให้เกิดโหมด Event-driven ได้หรือไม่

Private Sub txtRXTX_KeyPress (KeyAscii As Integer)

โพรซีเดอร์ txtRXTX_KeyPress(KeyAscii As Integer) ใช้ในการส่งข้อมูลไปที่ Com Port ซึ่งการส่งข้อมูลนั้นใช้ Output Property ดังตัวอย่างคือ MSComm1.Output = Chr$(KeyAscii) เป็นการส่งข้อมูล 1 อักขระให้กับ Serial Port

Private Sub MSComm1_OnComm()

โพรซีเดอร์ MSComm1_OnComm เป็นส่วนของการรับข้อมูลและ Events ต่าง ๆ ที่เกิดกับ Com Port ในโหมด Event-driven โดยการกำหนดให้ทำงานในโหมดดังกล่าว ซึ่งเป็นส่วนสำคัญหากต้องการทำงานในโหมดนี้จะต้องมีโพรซีเดอร์นี้

Public Static Sub ShowData (Term As Control, Data As String)

โพรซีเดอร์ ShowData(Term As Control, Data As String) นำข้อมูลใน Buffer ที่รับมาจาก Input Com มาแสดงที่ส่วนแสดงข้อมูลโดยตัวแปร Term As Control เป็นการส่งชื่อของ Control TextBox เพื่อให้โพรซีเดอร์รู้ว่าอ้างถึง Control อะไร ส่วน Data As String เป็นการส่งข้อมูลเข้ามาในโพรซีเดอร์เพื่อนำข้อมูลไปแสดง

Sub ที่กำหนดใน FrmProps.Frm ชื่อ FrmProperties

Private Sub cmdOK_Click()

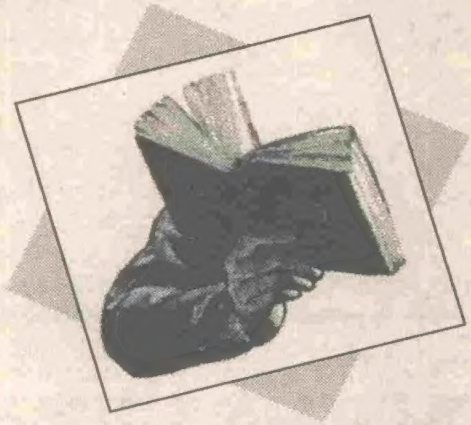
โพรซีเดอร์ cmdOK_Click เป็นการเก็บค่าที่กำหนด ซึ่งจะทำการเก็บลง Registry โดยการกำหนดค่าต่าง ๆ ใน SetCom ตัวอย่างเช่น การเก็บค่าลง Registry ของ Settings และ Com Port ดังนี้

- SaveSetting App.Title, "Properties", "Settings", frmVBCom.MSComm1.Settings บันทึก Settings ลงใน Registry ของระบบวินโดวส์
- SaveSetting App.Title, "Properties", "CommPort", frmVBCom.MSComm1.CommPort บันทึก CommPort ลงใน Registry ของระบบวินโดวส์

หลังจากได้รู้รายละเอียดโดยทั่วไปแล้ว ก็ถึงเวลาแล้วครับที่เราจะมาศึกษาการใช้งาน MSCOMM Control เพื่อติดต่อกับ Serial Port โดยการเขียนโปรแกรมเพื่อทดสอบการติดต่อกับ Serial Port โดยโปรแกรมจะแสดงการทำงานได้ทั้ง 2 แบบ คือ

1. การสื่อสารแบบกระตุ้นด้วยเหตุการณ์ (Event-driven) คือ การโปรแกรมที่ไม่ต้องรอการตรวจสอบข้อมูลเพราะเมื่อมีข้อมูลเข้ามาโปรแกรมจะไปทำงานในส่วนการรับข้อมูลโดยอัตโนมัติ (เกิดการอินเทอร์รัพต์ขึ้นเอง)

2. การสื่อสารแบบโพลลิ่ง (Polling) คือ โปรแกรมจะต้องวนรอบตรวจสอบข้อมูลตลอดเวลา



การใช้งานโปรแกรม VBCom

โปรแกรมตัวอย่าง VBCom นั้นจะมีหน้าตาดังรูปที่ 1 โดยสามารถเลือกโหมดการทำงานได้ 2 โหมด คือ

1. โหมด Event-driven คือ โหมดการทำงานแบบกระตุ้นด้วยเหตุการณ์

2. โหมด Polling คือ โหมดการทำงานแบบวนรอบรอข้อมูลตลอด

ซึ่งการทำงานก็เพียงแต่การคลิกเลือกโหมดเท่านั้น ส่วนปุ่มเลือกการทำงานนั้นประกอบไปด้วย 3 ปุ่ม คือ

1. ปุ่ม เปิด/ปิด พอร์ตการทำงานโดยการคลิกครั้งแรกจะเป็นการเปิดพอร์ต แล้วเมื่อคลิกอีกครั้งจะเป็นการปิดพอร์ตการทำงาน โดยดูได้จากปุ่มจะเป็น OnCom ก่อน จากนั้นคลิก 1 ครั้ง จะเปลี่ยนเป็น OffCom เมื่อคลิก OnCom แล้วจะแสดงสถานะการ OnCom ว่าใช้ Baud เท่าไร Data Bit, Stop Bit, Pararity Bit

2. ปุ่ม SetCom เป็นการกำหนดคุณสมบัติของ Com Port ในการใช้งาน แสดงดังรูปที่ 2

3. ปุ่ม Quit เป็นการเลิกการทำงานของโปรแกรม

การทดสอบโปรแกรม

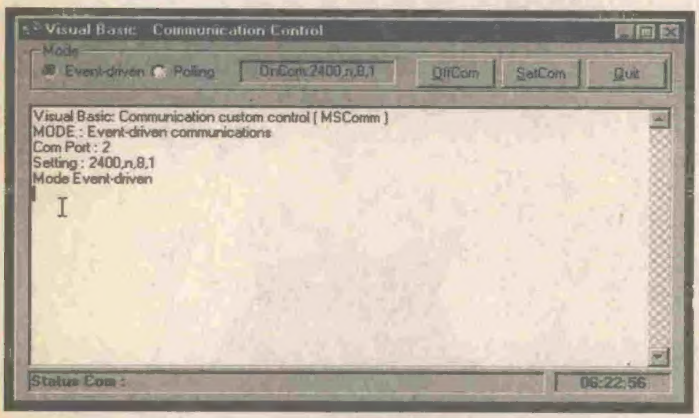
การทดสอบและใช้งานโปรแกรม VBCom ทำได้ดังนี้

1. ใช้คอมพิวเตอร์ 2 เครื่อง จากนั้นให้ต่อคอมพิวเตอร์ 2 เครื่องด้วยสาย RS-232

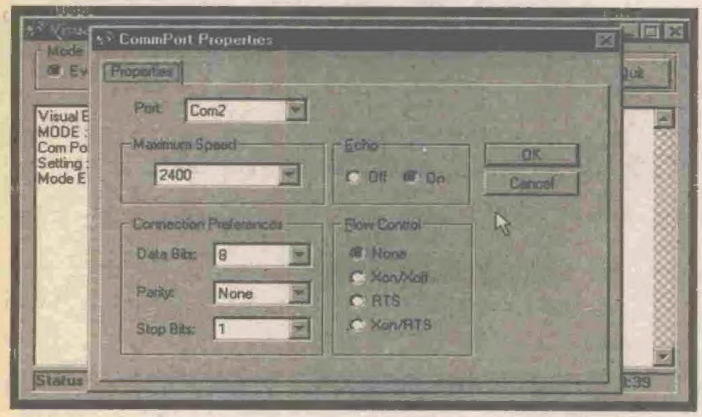
2. รันโปรแกรมทั้ง 2 เครื่อง กำหนดอัตราการติดต่อโดยการกำหนดที่ปุ่ม SetCom

3. กดปุ่ม OnCom โปรแกรมจะทำงานทันที

4. ที่ส่วนแสดงข้อมูล ป้อนข้อมูลแล้วจะไปแสดงที่อีกเครื่องทันที



รูปที่ 1 โปรแกรม VBCom



รูปที่ 2 การกำหนด Property Com Port

สรุปเนื้อหา

หลังจากได้ศึกษาการใช้งานโปรแกรมตัวอย่างแล้ว ผู้เขียนก็หวังว่าผู้เริ่มต้นการใช้งานโปรแกรม Visual Basic ในส่วนของการติดต่อกับ RS-232 คงพอจะมีแนวทางในการพัฒนาโปรแกรมมากขึ้น นอกจากนี้เรายังนำไปประยุกต์ใช้ในการทำงานด้านติดต่อกับอุปกรณ์ต่าง ๆ ที่มีการติดต่อกับอุปกรณ์ภายนอก โดยการผ่านทาง RS-232 ได้ ไม่ว่าจะเป็นเครื่องวัด เครื่องชั่ง เป็นต้น ที่มีการส่งข้อมูลผ่านทาง RS-232 ได้ หรือใช้คอมพิวเตอร์ในการควบคุมอุปกรณ์ต่าง ๆ ผ่านทาง RS-232 ก็ได้เช่นกัน

ท้ายบทความ

การใช้ Visual Basic สำหรับการเขียนโปรแกรมรับส่งข้อมูลผ่านทาง RS-232 จะเห็นได้ว่าการใช้งาน Control ของ Visual Basic นั้นไม่ได้เป็นเรื่องที่ยาก ซึ่งโปรแกรมตัวอย่างในบทความนี้ก็พอที่จะเป็นแนวทางในการศึกษาเพื่อที่จะนำไปพัฒนาได้ต่อไป โดยโปรแกรมนี้ผู้เขียนก็ได้ตัดทอนบางส่วนมาจากโปรแกรมตัวอย่างของ Visual Basic ซึ่งหากผู้ที่สนใจต้องการซอร์สโค้ดก็สามารถติดต่อขอใช้บริการได้จากซอฟต์เมล์ สวัสดีครับ

ดับเบิลคลิก ทางลัดที่ควรหัดใช้

การสั่งงานสำหรับโปรแกรมบนวินโดวส์ บางอย่างมีการกำหนด Button บนทูลบาร์ไว้ ทำให้สะดวกต่อการใช้งานมากขึ้น บางอย่างกำหนดคำสั่งที่เรียกว่า Shortcut key ไว้ แต่ Shortcut จำเป็นต้องจดจำ ทำให้ยุ่งยากพอสมควร แต่ถ้างานไหนไม่ได้กำหนดไว้จะต้องเรียกผ่านเมนูอย่างเดียวบางครั้งก็ยุ่งยาก ทางลัดหนึ่งที่ไม่ค่อยพบเห็นคนเลือกใช้คือ วิธีการ ดับเบิลคลิก ซึ่งผมขอยกตัวอย่างให้ทราบดังนี้

ในโปรแกรม MS-WORD กรณีเอกสารมีหลายหน้า และต้องการเคลื่อนย้ายไปหน้าอื่นซึ่งอยู่ห่างไปอีกหลายหน้า คุณเพียงแต่นำเมาส์เคอร์เซอร์ไปที่ Status bar บริเวณที่มีคำว่า Page Sec หรือ At Ln Col จากนั้นดับเบิลคลิกจะปรากฏไดอะล็อกบ็อกซ์ Go To ขึ้นมาให้กำหนดค่าหมายเลขหน้าที่ต้องการย้ายไป ง่ายกว่าการใช้เมาส์คลิกไปทีละหน้าไปเรื่อย ๆ

ในโปรแกรม MS-EXCEL ถ้าคุณพิมพ์ข้อความที่ยาวกว่าความกว้างของคอลัมน์ คุณสามารถปรับให้คอลัมน์กว้างพอดีกับข้อความได้โดยการเลื่อนเมาส์ไปที่ Column header จากนั้นดับเบิลคลิกที่ตำแหน่งคั่นระหว่างคอลัมน์ โปรแกรมจะปรับให้ความกว้างคอลัมน์พอดีกับข้อความที่ยาวที่สุดทันที ง่ายกว่าการใช้เมาส์ค่อย ๆ ปรับ เพราะบางครั้งข้อความที่ยาวนั้นอยู่ด้านล่าง ซึ่งคุณจะต้องเลื่อนมาที่ข้อความที่ยาวที่สุดก่อนจึงจะเห็นว่าความกว้างเหมาะสมหรือไม่ ซึ่งยุ่งยากมาก เพราะอาจจะมีข้อความที่ยาวกว่านั้นอีกก็ได้

หวังว่าวิธีดับเบิลคลิกนี้จะช่วยให้คุณผู้อ่านสามารถใช้งาน MS-OFFICE ได้สะดวกและรวดเร็วมากขึ้น ลองดับเบิลคลิกที่ตำแหน่งอื่นบ้างนะครับ คุณจะได้ทางลัดอีกมากมายที่คุณอาจคาดไม่ถึงก็ได้ ซึ่งทางลัดนี้ใช้ได้ไม่เหมือนทางลัดในกรุงเทพฯ หรอกนะ สวัสดีครับ

เจ้าของเทคนิค
คุณดุสิต เลาหสินณรงค์
43/4 ถ.กำแพงเมือง ต.ท่าราบ อ.เมือง จ.เพชรบุรี 76000
(เจ้าของเทคนิคจะได้รับหนังสือ เทคนิคการเดินสายระบบเครือข่ายคอมพิวเตอร์ จำนวน 1 เล่ม)